AN ARGUS SPECIALIST PUBLICATION

November 1985

£1.50

A&B COMPUTING

FOR USERS OF THE BBC MICRO AND ELECTRON

In the Beginning was the Word

Software

- Interactive Printer Character Design
- Adventurescape Text Editor
- BASIC utilities

Full feature machine code wordprocessor for BBC and Electron

Amnesia, 100K of Adventure challenge

On offer–

Acornsoft's Talkback, ABC Reviewed

List Processing – Logo, children and language

Electronic Office – Ergonomics

Notes on MIDI – Powertran, Siel

BBC PLUS 128K INSIDE ▷▷▷ BBC PLUS 128K INSIDE ▷▷▷ BBC PLUS 128K INSIDE

A&

Volume Three Number

Editor: Mark Webb
Assistant Editor: Fiona J.Eldridge
Group Editor: Wendy J.Palmer
Advertisement Manager: Jonathan McGarry
Editorial Director: Ron Moulton
Managing Director: Peter Welham

REGULARS

FEATURES

Articles in the next month column are in an advanced state of preparation but cannot be guaranteed to appear.

A&B Computing is constantly on the look-out for original and well-written articles and programs for publication. Feel free to submit your work to us for consideration for publication.

All submitted material must be in machine readable form. This applies both to programs (in any language) and to documentation, which should be prepared with a BBC or Electron wordprocessor. 5¼ inch disc (40/80) or cassette equally acceptable. Please also include hardcopy and any suitable illustration, photographs and/or screen dumps.

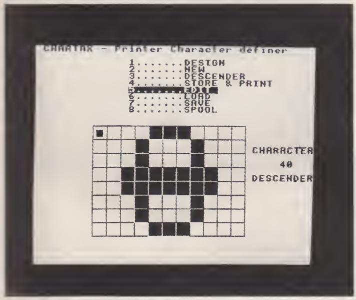
If you are considering submitting material to A&B then please send a S.A.E. for a comprehensive style sheet. It's also sensible to give us a ring before going ahead with any major work.

All submissions will be acknowledged and material returned if not required. On acceptance the copyright in such works which will pass to Argus Specialist Publications Limited will be paid for at competitive rates. All work for consideration should be sent to the Editor at the above address.

Eleven November 1985

Published by Argus Specialist Publications Ltd., Number One, Golden Square, London W1R 3AB. Tel: 01 437 0626.

All work for consideration should be sent to the Editor of A&B Computing at our Golden Square address.



```
CHARTAX - Printer Character definer
          1.......DESIGN
          2.......NEW
          3.......DESCENDER
          4.......STORE & PRINT
          5.......EDIT
          6.......LOAD
          7.......SAVE
          8.......SPOOL

                                 CHARACTER
                                    40
                                 DESCENDER
```

REVIEWS

SOFTWARE

A&B Computing is published monthly on the first Friday of the month preceding cover date. Distributed by SM Distribution Ltd., 16-18 Trinity Gardens, London SW9 8DX. Telephone: 01 274 8611. Printed in the UK by Garnett Print, Rotherham and London.

Printed in the UK by Garnett Print, Rotherham and London.

BBC+ 128K

Gordon Taylor

What It Is and What It Does

The B Plus 128K provides an additional 64K of RAM (in just two 64 x 4 chips) on a small board which is mounted on the expansion connector on the right-hand edge of the main board.

With appropriate software, this can be used in any of three ways:-
- As four 16K (or two 32K) banks of Paged (or Sideways) RAM. This may be loaded with images of ROM software that then behave exactly as though the ROMs were plugged into the machine (until it is switched off !).
- As a RAM disc of 64K.
- As a Virtual Address Space of 64K above the 64K that is directly addressable by the 8-bit Central Processor Unit (CPU). (See Fig. 1).

It could also be shared (e.g. in 16K blocks) between these three uses, again with appropriate software.

Software to load and save Paged or Sideways RAM banks (but not to save ROMs directly), and to move data from main RAM to Paged RAM, has been added to the B Plus DFS ROM — which thus becomes version 2.2. However, software to use the expansion RAM as a RAM disc (as in the Atari 130 XE, for example), is not planned by Acorn but is likely to be offerred by others.

Indeed, the Paged or Sideways RAM and RAM disc uses may be compared with those of the Solidisk Sideways Systems in a standard Beeb. (see A & B, November 1984, p 66).

In order to use the Virtual Address Space between &10000 and &1FFFF, the "language" must include code to translate this to a corresponding bank number and an address between &8000 and &BFFF. For example, addresses between &18000 and &1BFFFF are translated to bank C. This is provided in BAS128, which (like BASIC 1 and BASIC 2) can address only 64K, but (at some 19K) is larger than one bank, and is therefore loaded from disc into main user RAM.

```
                        HOW THE B PLUS 128 K WORKS

       MEMORY MAP (SIMPLIFIED) WITH BANK NUMBERS AND ADDRESSES IN HEX

                                                     - - - - ___
                                                    .       ¦ V ¦     &1FFFF
                                                    .BANK D ¦ I ¦
                                                    .       ¦ R ¦
                                                    .       ¦ T ¦     &1C000
                                              - - - - - - - ¦---¦
                                                .   .   .   ¦ U ¦
                                                .  BANK C   ¦ A ¦
                                                .   .   .   ¦ L ¦
                                                .           ¦   ¦     &18000
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - ¦___¦
    .                                           .   .   .   ¦ A ¦
    .               BANK 1                      .   .   .   ¦ D ¦
    .                                           .   .   .   ¦ D ¦
    .                                           .   .   .   ¦ R ¦     &14000
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ¦---¦
.   .   .                                       .   .   .   ¦   ¦
.                   BANK 0                      .   .   .   ¦ S ¦
.   .   .                                       .   .   .   ¦ P ¦
.                                               .   .   .   ¦ C ¦     &10000
.   - - - - - - - - - - - - - - - - - - - - - - - - - - - - ¦---¦
.   .   .                                       .   .   .   ¦MOS¦      &FFFF
.   .   .                                       .   .   .   ¦ROM¦
.   .   .                                       .   .   .   ¦   ¦
.   .   .                                       .   .   .   ¦   ¦      &C000
¦EXP¦EXP¦  F¦OUR¦ 32¦K R¦OM ¦SOC¦KET¦S  ¦DFS¦DFS¦EXP¦EXP¦BAS¦BAS¦
¦RAM¦RAM¦   ¦   ¦   ¦   ¦   ¦   ¦   ¦   ¦ROM¦ROM¦RAM¦RAM¦ROM¦ROM¦RAM¦ &AFFF
¦ 0 ¦ 1 ¦ 2 / 3 ¦ 4 / 5 ¦ 6 / 7 ¦ 8 / 9 ¦ A / B ¦ C ¦ D ¦ E / F ¦ 80¦
¦___¦___¦___¦___¦___¦___¦___¦___¦___¦___¦___¦___¦___¦___¦___¦___¦___¦ &8000
                                                            ¦   ¦SCR¦
                                                            ¦USR¦EEN¦
                                                            ¦   ¦RAM¦
                                                            ¦   ¦___¦ &3000
                                                            ¦RAM¦
                                                            ¦___¦      &0000
```

Thus the figures for the Acorn machines are :-

Computer	Size for BASIC — K	Average Time — Secs
B Plus with BASIC 2	25.5	14.55
B Plus with BAS128	64	29.43
B Plus + 6502 with BASIC 2	44	9.83
Electron with BASIC 2	20.5	20.44

While those for the main rivals are :-

Computer	Size for BASIC — K	Average Time — Secs
Amstrad 6128	41.3	N/A
(Amstrad 464	45.5	14.59)
Atari 130 XE	40	75.5
Commodore C 128	37.5	N/A
(Commodore C 64	37.5	34.0)
Sinclair Spectrum 48K	40	58.5
Sony (and other) MSX	32 (26 with disc)	44.3

(The figures for the rival machines are taken from results published by Personal Computer World and Practical Computing).

MEASURES

Two measures of a computer are the size available to BASIC (and other) programs and the speed. One purpose of expanding the B Plus to 128K is to allow it to run larger programs, and it has 64K available to BASIC when using BAS128. Speed in BASIC is often expressed as the average time to carry out the eight Personal Computer World BASIC Benchmarks. While this is just twice as long as when running BASIC 2, it is still shorter than most of its rivals. A powerful combination of size and speed is still available by using a 6502 Co-Processor, running Hi-BASIC.

VALUE

The value of the expansion to 128K could be even greater for other "languages", including word processors etc. Regarding these, Acorn will only say that "versions of other software will be developed". One may however speculate that :-
- this will include View, since with up to 25.5K in main memory and 64K in the Virtual Address Space, this could offer up to 89.5K — i.e. over 20 pages of A4 typed in single spacing. (This compares with "only" 47K with Hi-View running in a 6502 Co-Processor).
- this will not include ViewSheet, which uses 15-bit pointers for speed, and so is limited to a maximum sheet size of 32K (unless completely re-written). (It can of course handle much larger "models" by consolidating the results of multiple sheets).

The 64K RAM expansion is to be available both as an upgrade for existing B Plus machines, and already built-in to new machines. It includes the expansion RAM, the new DFS 2.2 ROM, BAS128 on disc and an explanatory leaflet. The price of the upgrade is likely to be only £30 plus fitting. This should be done by a dealer, since it involves soldering a multi-pin connector to the main board, and would still be excellent value at say £10. The price of the already-expanded B Plus 128K is £499, including VAT — the same as for the 64K model until recently. Taken together, these developments amount to a significant increase in price/performance, which — with the ever improving software — will help to keep the Beeb competitive for a considerable time to come. ■

Making the most of LOGO

John Henderson

Starting list processing with Logotron.

Learn to live and live to learn. This motto must surely apply to the use of LOGO in schools. However many ideas users come up with, there are more possibilities around the corner. Using a full version of LOGO enables children of all ages to experiment with problems. Trying and testing solutions is a natural part of learning.

To date, most LOGO applications have concerned themselves with Turtle Graphics. A much wider application is to use the power of list processing to assist learning. This means much more than analysing the weekly shopping list as you will begin to see from this article. List processing enables the writing of procedures to explore the way words fit together. That is not to say that it follows on naturally from turtle graphics. List processing is very different as it does not build on concepts that are immediately familiar to children, nor does it provide instant solutions. A great deal of time and effort are required to solve problems.

LOGOTRON

All the procedures used in this article were developed using **LOGOTRON.** This was chosen for three reasons. First, it is a full implementation of LOGO, and thus provided the extension facilities that schools are looking for. Second, it seemed the most versatile and speedy package of the Big Four LOGOs for the job of list processing on the BBC computer. Third, it is contained on a single chip and eliminated the need for an expansion board.

The LOGOTRON package consists of a full manual of commands and sample programs, a command summary sheet and one 16K chip. Although there are several spelling mistakes within this manual, presentation is pleasing with actual program lists highlighted in red. The ring binder containing the file allows room for supplementary sheets to be added. The section on Logo Grammar is particularly clear and helpful. LOGOTRON was written by ACT/SOLI to LCSI standard.

LISTS GALORE

LOGOTRON takes lists of words, which are referred to as **OBJECTS,** and processes them. Easy isn't it ! Well I hope it will be clearer by the time you reach the end. LOGO works through a series of commands called primitives.

```
FORWARD 200
LEFT 90
```

are commands used often in turtle graphic programs. Not all primitives are commands — some are actual outputs termed operations.

In list processing three types of data objects are handled. Objects can be a words, numbers or a list. They can then be used as input/output from procedures.

```
MAKE “1 [HELLO]
PRINT :1
HELLO

MAKE “2 365
PRINT :2
365
```

This is an example of printing a word or a number from a single variable command. Square brackets can be eliminated from numbers. It is easy to join words together using the **SENTENCE** command.

```
MAKE “1 [ELEMENTARY MY]
MAKE “2 [DEAR ...]
PRINT SE :1 :2
ELEMENTARY MY DEAR ...
```

Notice the use of SE as an abbreviation and the use of a colon before each part (each element) of the list. If the sentence to be joined consists of more than two elements, brackets must be used as in the example below:

```
MAKE “3 [WHATS YOUR NAME ?]
PRINT (SE :1 :2 :3)
ELEMENTARY MY DEAR ... WHAT'S YOUR NAME ?
```

PRINT statements can be more powerful when combined with other commands. Look at the variable name **LABEL** given the object LOGOTRON:

```
MAKE “LABEL “LOGOTRON
```

By adding commands to this, different displays result.

```
PRINT FIRST :LABEL
```

prints the first element of the object that is LABEL, hence on screen appears:

L

Similarly, printing the last element of LABEL needs the command LAST:

```
PRINT LAST :LABEL
N
```

Commands **BF** (everything BUT LAST element) and **BF** (everything BUT FIRST element) yield more power with words:

```
PRINT BL :LABEL
LOGOTRO
```

How can children use these commands ? Well letter patterns are important to children learning to spell, so the

LOGO LISTING 1

shows how to strip a word a letter at a time and print out the result. Children can use these patterns to learn how to look at the shape of word strings.

```
MAKE “LIST [SQUARE TRIANGLE RECTANGLE]
PRINT LAST :LIST
RECTANGLE
PRINT BL :LIST
SQUARE TRIANGLE
```

Some other commands used by LOGOTRON make writing on the computer a painless activity. Printing to a certain part of the screen, for example, is made simple using the command **SETCURSOR (SC):**

```
SETCURSOR [15 15]
```

sets the printing position approximately in the middle of the screen.

Commands **READLIST (RL)** and **MEMBER?** are versatile and powerful. The list below shows a procedure to... well you look:

```
TYPE [WORD ?] MAKE “WORD RL
```

PROGRAM LISTING

LISTING 1 — WORD STRIP:

```
TO PAT :LETTER
PRINT :LETTER
PAT BF :LETTER
END

WORD :LOGOTRON

LOGOTRON
OGOTRON
GOTRON
OTRON
TRON
RON
ON
N
```

Compare this to the BASIC version of the same program:

```
10 X$=LOGOTRON
20 FOR loop =1 TO 8
30 PRINT LEFT$ (X$,loop)
40 NEXT loop
```

```
Word? A & B
PRINT :WORD
A & B
```

The command MEMBER? does the important job of checking outputs to see if they are true. This is especially useful to check if an object is included on a list:

```
MAKE “LIST1 [A & B]
MAKE “LIST2 [computing]
IF MEMBER? “computing :LIST2 [PRINT ”Correct]
```

CONTINUED OVER

APPLICATIONS

Look at ***LOGO LISTING 2.***

This forms the basis of a word puzzle created by children for use in the classroom. The advantages of this system are that words can be included that are *DIRECTLY* related to the work in a particular classroom and any redundant words can be removed, thus giving more relevance to the situation. The puzzle is based on three main procedures. First, a datafile containing the words required, second a program to ensure that the correct matches are made, and third a random choice element to the list. Creating a series of useful utilities such as this can free many teachers from the task of writing their own programs to being able to concentrate on constructing learning materials for children.

LISTS FOR CHILDREN

It may be that children will use the list processing facilities to learn how to program. Certainly there are some user-friendly aspects to list processing which are nearer to our native language than their BASIC equivalent. LOGO listing is not restricted in the same way as BASIC. Lists can be of varying lengths (up to 255 characters), may be completely empty or may contain sublists within them. However the vexed question of whether to teach children to program has not disappeared. Educationally, effort should be put into problem-solving rather than being able to write programs to overcome simpler problems.

Why LOGOTRON ? It's list processing speed is the fastest of the Big Four LOGOs. In a test using a 10 element list printed out 100 times the clock averaged 4.2 seconds. Error messages try to be helpful, although inevitably the user guide is required at first to interpret some of these. There is also sufficient workspace, the amount of memory available to the user, to build procedures that children are likely to want. Storing procedures in the Editor allows more space to the user, although the best policy is to train children to SAVE procedures onto disc as they work. LOGOTRON incorporates a garbage collection system to release workspace.

Table One indicates the size of workspaces available. Finally, the range of commands available is sufficient to allow schools access to the power of lists.

Table Two contains the most useful commands that children were introduced to when starting list processing with LOGOTRON.

THE FUTURE

Without a doubt the computer is here to stay. Without a doubt LOGO will be a powerful learning tool in the classrooms of the near and distant future. The question is what will be expected to happen ? Many primary schools and some of the children in them, will be quite stretched to learn in a turtle graphics microworld. It is very difficult to use lists with confidence. There are, however, ways around the problem by using toolbox programs, written by teachers with particular children in mind.

Content-free applications that children can adapt to their own needs at that time without being driven to distraction trying to battle with the full power of list processing. To push children, and teachers, into microworlds where they lose control will be to tempt fate — even if it is high technology.

The most educationally beneficial situations in which children find themselves allow a degree of freedom, coupled with support from the teacher and time to achieve results, in an effort to solve problems. It is highly likely that schools will concentrate on a few of the many facilities available through LOGO. It is for individuals to choose according to their situation. Using LOGO in musical and control microworlds has yet to be explored. Today the emphasis is shifting to applications for the power of list processing eg. creating a database facility using LOGO and enabling children to really explore the world of words where they are in control.

TABLE TWO: LOGOTRON LIST PRIMITIVES:

BUTFIRST (BF)print out all but the first element of object

BUTLAST (BL)print out all but last element of object

COUNT outputs the number of elements in an object

FIRST output first element of object

FIRSTPUT (FPUT) places object at start of a new list

ITEM outputs a specific item from a list

LAST output last element of object

LASTPUT (LPUT)places object at end of a new list

MAKE creates a variable

MEMBER? checks the TRUE value of an object

SENTENCE (SE)creates a list from two or more inputs

TABLE ONE: LOGOTRON WORKSPACE

This varies with the screen MODE being used, MODE 4 being the default mode. Space is measured in NODES, each node being approx. 5 bytes.

MODE 0	755
MODE 1	755
MODE 2	755
MODE 3	1571
MODE 4	2795
MODE 5	2795
MODE 6	3203
MODE 7	4631

PROGRAM LISTIN

LISTING 2 — WORD LISTS

```
TO DATA
MAKE "LIST1 [compu
monitor disc]
MAKE "LIST2 [casset
discs lists]
END

TO MATCH
DATA
SETMODE 7
REPEAT 3 [PRINT ( SE
CHOOSE :LIST1 [is alw
hardware whereas] CHOOSE
:LIST2 [is always software]
END

TO CHOOSE :LIST
OP ITEM 1 + RANDOM
:LIST
END
```

LOGOTRON LOGO
Logotronics
Regional Office:
5 Granby Street
LOUGHBOROUGH
LEICS.
LE 11 3DU

£59.95

Making the most of Assembler

Margaret Stanger

Sometimes machine code is so fast that a routine has to be written to slow it down while the rest of the world catches up!

The short program WAITING has a short BASIC delay routine, and the equivalent in machine code. The TIME is set to MINUS the chosen interval, and the computer does not continue the program until the TIME reaches zero.

The chosen interval (5 seconds in this case) was stored from location &80 to &84, with &80 containing the hundredths of a second.

ie. !&80=-500:?&84=&FF.

LOCATION	VALUE	VALUE
&80	12	&0C
&81	254	&FE
&82	255	&FF
&83	255	&FF
&84	255	&FF

The time could then be written using the OSWORD 2 routine as described in the 'User Guide'. The accumulator is set to 2 and the X and Y registers point to a parameter block starting at &0080.

The TIME is read using the OSWORD 1 routine, but this time the parameter block starts at &0070. Time marches on and the changes are reflected in the following locations after 2.43 seconds and 5 seconds:-

LOCATION	value1	value2	value3
&70	12	0	0
&71	254	255	0
&72	255	255	0
&73	255	255	0
&74	255	255	0

The computer checks location &74 for the value zero, before the program continues with a return to BASIC.For a BBC machine, the interval timers could be written and read in a similar way using OSWORD 4 and OSWORD 3 (See Micro Eventing next month).

Taking time out in Assembler

```
 10REM**WAITING BY M.STANGER
 (5FBF)
 20DELAY=500:REM 5 SECONDS
 (068A)
 30PROCAS (2536)
 40!&80=-DELAY:?&84=&FF:
 CALL&A00 (B7B7)
 50VDU7:END (54E0)
 60: (0C0C)
 70DEFPROCCHEAT (5BD8)
 80TIME=-DELAY (F528)
 90REPEATUNTILTIME=0 (B96F)
100ENDPROC (8486)
110: (63DA)
120DEFPROCAS (7522)
130FORPASS=1TO2 (7933)
140P%=&A00 (9801)
150[OPT PASS (1516)
160LDX#&80 (2CC3)
170LDY#0 (7CB8)
180LDA#2 (398A)
190JSR &FFF1\OSWORD (BC1D)
200.STAY LDX#&70 (3C7B)
210LDY#0 (77EE)
220LDA#1 (92C2)
230JSR &FFF1\OSWORD (1B52)
240LDA&74 (D8BA)
250BNE STAY (2A1B)
260RTS (6270)
270] (C2D8)
280NEXT:ENDPROC (DC75)
```

VARIABLES

DELAY chosen interval
PASS loop variable for two pass assembly

MAIN PROGRAM

line
20 Sets DELAY to 5 seconds
30 Assembles machine code
40 Stores DELAY in parameter block, calls Machine Code routine.
50 Ends the program after sound signal

PROCEDURES

PROCCHEAT
This procedure contains the equivalent of line 40 in BASIC.
90 Sets TIME to MINUS the delay
100 Waits until the TIME reaches zero

PROCAS
130 Two pass assembly
140 Sets the program pointer or start of the Machine Code to &A00, as this area is usually spare
150 Sets assembly options
160 Loads X register with Least Significant Byte of parameter block
170 Loads Y register with Most Significant Byte of parameter block
180 Loads the accumulator with 2
190 Jumps to OSWORD subroutine at &FFF1 to write the time
200 Loads X register with LSB of parameter block
210 Loads Y register with MSB of parameter block
220 Loads accumulator with 1
230 Jumps to OSWORD subroutine at &FFF1 to read the time
240 Loads the accumulator with &74
250 Contines program if accumulator is zero, otherwise repeats from LOOP
260 Returns to BASIC
270 End of machine code
280 Next pass of assembler ■

Notes on MIDI –

Clive Grace

In the first part of a series, some of the best music making hardware and software comes under scrutiny.

As there are more and more computer users and potentially more computer musicians, so the increasing importance of the computer is beginning to show in today's contemporary and popular music.

Look at Duran Duran's "Reflex" video, and you will see Nick Rhodes, their keyboards player at the controls of two £27,000 Fairlight sampling computers (more about that later). The Thompson Twins brought one on stage in the American Live Aid concerts, Johnny Fingers, from the Boomtown Rats had an Apple-based Greengate DS3 sampling unit on stage as his main keyboard.

In the April and May issues of A&B, I took the Acorn Music 500 synthesiser into a studio to try out its facilities. Music 500 is geared towards being a home synthesiser, and thus a lot of potential studio owners will not have heard of AMPLE, the best music composition language I have yet seen, but many studio engineers and computer musicians will want a great deal more. Sound sampling is all the rage at the moment. MIDI is an absolute must if you wish to use more than one synthesiser or computer together, and some sequencing software is necessary if you want to build up songs or patterns which hold together many of todays pop songs.

WHAT IS SOUND SAMPLING AND MIDI?

Sound sampling as opposed to Sound Synthesis, is a process by which a sound is stored digitally within a computer, it can then be re-calculated and the waveform altered to play at different pitches, so the original sound can be played back in a musical score. The advantage that sampling has over synthesis is quite obvious. A synthesised sound will never be quite right as it is based upon parameters using a number of oscillators and filters to distort a pure wave form, like a sine wave or a square wave. A harpsichord will never sound quite right as the sound wave is so complicated. To the discerning ear, the resultant sound would be annoying to say the least.

The obvious alternative is to make an audio copy of the original instrument and, by a system of digitising the sound, (normally via a pulse code modulation process) storing the sound in the computer and eventually onto a disk. So if you wanted to play one of Purcell's Harpsichord suites on an original instrument, perhaps sampling a 1705 Cawton Aston spinet would do the job, unethical I suppose, but safer than shipping a priceless (and beautiful) instrument on tour.

The most famous sampling system is the CMI Fairlight, which has an awesome array of sound processing software and a friendly "human interface" via a lightpen. The Fairlight costs an absolute packet (as high as £32,000 for the top version) and boasts some of the most detailed software ever written on a computer.

MIDI is rapidly becoming a communications standard in the world of digital musical instruments, this can include anything from a Roland Synthesiser to a mixing desk controller unit. If an instrument does not support MIDI then it will not be able to synchronise itself with other MIDI instruments, nor would you be able to record a piece of music in the studio unless it was a live overdub. MIDI is an important step forward in that (grumbles from Bob Moog aside) it allows the musician to effectively control the flow of data from one digital instrument to another and is in reality a communications standard.

POWERTRAN SOUND SAMPLING UNIT

One of the most outstanding pieces of hardware to come out for the BBC Micro this year is the Powertran MCS-1 sound sampling unit. It is a high quality digital sound sampling unit with a digital delay to boot. It is a professional unit which has found users in many fields of music. Midge Ure from Ultravox is a well known user of the MCS-1.

The inclusion of a digital delay unit is, in my opinion, a very good one. Many sounds can be "thickened up" by adding a small delay (of up to 200 Milliseconds) thus giving the impression that there is more than one sound source. Another useful trick is to have a shorter delay on parts of a waveform like a Fretless Electric Bass, to give the sound a rubbery String Bass sound. This process is called *Flanging* and can be heard in the bass guitar on Paul Young's albums (or even better, John Martyn's "Well Kept Secret" album).

The unit is quite heavy and is designed to be rack mounted into a travelling framework or in the studio. It has an alarming number of switches on the front and a healthy number of interface ports on the back.

FEATURES

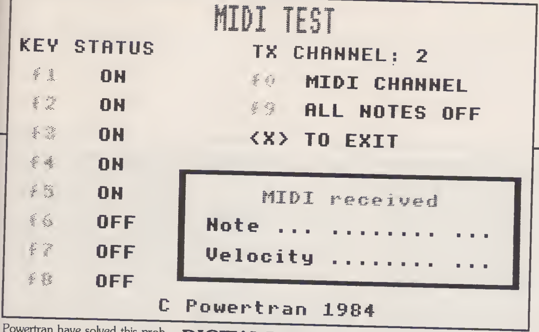
The MCS-1 has an impressive array of features which are all addressed from the panel of switches to the right of the digital display. As the sampling unit is MIDI controllable the sampled sound is controllable from any MIDI equipped keyboard or by special modification, a 1 Volt per octave keyboard like ATPL's Symphony keyboard or the Music 500 keyboard.

Sampling is a very simple process by which the sound is fed into the MCS-1 via either a microphone input, or a pure sound source, which is useful if you want to rip-off all of the best sounds on a synthesiser which you have borrowed for the weekend. The microphone or direct inject plugs into the socket on the far left of the unit marked "IN". Connection to a mixing desk or amplifier is made via the "OUT" socket, mid left of the display.

Pressing the button marked "RECORD" starts a pulsing LED on the button, this is to tell the user that it is ready to record a sound. The unit will store the sound, (up to approximately 4 seconds) upon a sound loud enough to trigger the record facility, or manually by pressing the record button again. The total length of time sampled can be reduced, so as to cut out any unwanted sections of a sample. Both the RAM size button and the sample speed controls are used to adjust the length of playback sample.

Thankfully, Powertran have included a "LEVEL" control with a "CLIP" LED below it, this is essential as you want a signal to be as loud as possible without distorting the incoming signal to sample or delay. Should you wish to protect the sound you are playing with (before dumping it down into the BBC software) you may wish to use the "FREEZE" option. This proved useful when I was sampling a Violin solo from Elgar's Violin Concerto, and had found the perfect sample of a Violin thanks to Itzhak Perlman. Pressing "FREEZE" saved the day as it avoided a near fatal mistake (to the sound) when I tried to sample another sound before commiting the Violin to disk.

There is a noise reduction facility built into the MCS-1 which works via a de-quantization process. Normally when recording I trust the experience of the Engineer who is either a DBX or Dolbyphile. The NR button is set "on" at power up, and is invaluable when playing live, which is normally an appallingly noisy affair, if you fall foul of a bad PA system, or by trying to run 10 keyboards from three 4 gang sockets. The less noise that has to be clipped off at the mixing desk, the better and the NR facility goes a long way to curing this. Quantization noise from the sound being stored is sadly inevitable and



```
                    MIDI TEST
KEY STATUS               TX CHANNEL: 2
  f1    ON               f0   MIDI CHANNEL
  f2    ON               f9   ALL NOTES OFF
  f3    ON               <X> TO EXIT
  f4    ON
  f5    ON                  MIDI received
  f6    OFF          Note ... ......... ...
  f7    OFF          Velocity ........ ...
  f8    OFF
                C Powertran 1984
```

Powertran have solved this problem. However should you require the additional brightness, then you can set this switch off, but it is not advisable.

The "PLAY" button is simply a means in which the sound can be recalled in its raw sampled form, connecting a microphone and adjusting the level control can, in a fit of self indulgence produce effects *exactly* the same as those found in Paul Harcastle's "19" or Eurythmic's "Sexcrime", ie the heavily sampled sounds, repeated over and over again, before the sample is actually finished..... fun for a while and proves that one of the most current effects in modern popular music requires no talent at all! Using the play button in this way is of no professional use. "PLAY" is used to examine the pure sample, and to modify it without the use of a MIDI keyboard, so sounds that require looping, especially the Violin which requires an extended delay or a Piano which needs an unnaturally extended decay, as the amount of sampling would render the MCS-1 too expensive in RAM chips.

"VOICE MODE" is an essential feature for playing the sound on a MIDI keyboard. whereby the pitch and duration of the sound (a maximum of 5 octaves) can be controlled, including the "bend" key to allow extra expression in keyboard solos. This is also a requirement when using "GATE" mode, the sound is only on for the set gate sound, this is useful for sounds which need to be clipped or, need a quicker decay. Pressing this button causes a "Lo" or a "Sh" prompt on the display which selects the long or short gate delay, the GATE mode allows for a great deal of control over loop effects.

DIGITAL DELAY EFFECTS

A special button for the delay line operation is set aside, and with the "SWEEP ON-OFF" buttons, a huge array of delay effects are possible, from 0.3 seconds delay (which is like singing into a tin can) to a maximum of 32 seconds (which is like singing in the Alps). Constant regeneration effects, such as the ones found in Reggae "Dub" music and, to more experimental effect by Robert Fripp, are possible, by making sure that the sound is re-echoed. Turning off the noise reduction leads to dynamic semi-distorted echo effects which is a similar sound to the echo effect on the guitar in Genesis' *Second Home By the Sea* .

The sweep buttons are used to affect the level of Phasing and Flanging, heavy sweeps are usually required on bass sounds, whereas string sounds are less demanding, pure vibrato is possible by setting the sweep range and sweep speed to maximum. Vibrato is a nice effect if used properly, and can add depth to a sound. The vibrato effects on the old Moog Prodigy and Rouge synthesisers are a case in point.

The rear panel is however a little easier to explain, as most of the connections are here. The connections are normally made and forgotten about. In the MCS-1's case there is only the provision for a MIDI IN and OUT socket, there is no MIDI THRU which is a shame as this means that the MCS-1 will have to be situated at the end of a particular MIDI node (if a satellite MIDI arrangement is employed) It is getting a little worrying how so many synthesiser and drum unit producers are not fully implementing the MIDI THRU as this allows the MIDI information to be passed through the unit and onto another piece of equipment on another MIDI channel, of which there are eight in total.

There is a "FREEZE" socket which is used in conjunction with a short circuiting footswitch and is best reserved for freezing a sound, especially if you are working out an echo effect and have to have both hands on the MCS-1, the foot switch effectively keeps "PLAY" or a similar keyboard operation going.

The main link between the BBC and the MCS-1 is not through the MIDI interface, but by a special BBC IN-OUT socket, a parallel 180 degree DIN plug is all that is needed to connect the BBC to the MCS-1 and thus control the unit via the software supplied on disk.

BBC MIDI INTERFACE

Connecting the MCS-1 to the BBC Micro is very simple, the Powertran MIDI interface plugs into the 1MHz Bus underneath the BBC and the MCS-1 is connected to the interface by the BBC IN-OUT to the socket marked MCS1.

The MIDI interface is very simple to use and can be powered by the MCS-1's generous power supply or by an additional 9V (100mA) power supply. There are two LEDs on the unit, on the left is the power LED, which is to inform the user that the interface is connected and working properly. On the right is the data transfer LED which is only alight when data is being passed from the MCS-1 socket or the MIDI IN connector. The sound travels down the connectors at just over 31KHz to the BBC which is the accepted fast MIDI transfer mode, although the data is most definitely *not* MIDI information.

The unit is very small, just 150mm wide and 50mm high, the unit is unobtrusive and very utilitarian, all of the cables are easy to get at, which is necessary, especially as I have got into the habit of physically changing MIDI cables instead of re-allocating MIDI channels.

The Powertran MIDI interface differs from other MIDI interfaces in that it can control the MCS-1, no other MIDI interface is able to do this, so if you wish to store sounds on the BBC and use MIDI instruments as well, then you will have to use the Powertran interface.

CONTROL SOFTWARE

The controlling software is vital, if you want to do anything at all with the MCS-1 and the MIDI interface, and as the MCS-1 data is not a MIDI format it requires a separate transmission bus.

Each sound on the MCS-1 occupies 64K bytes of RAM and thus a complete sample of data will take about 21 seconds to transfer onto the BBC disc, not immediate, but fast enough for many practical purposes between songs. Unlike the Greengate DS3 with which you can store a number of samples in the machine at one time (providing they are short, like drums up to four decent samples can be squeezed in). The Powertran unit is only capable of storing one sound. No provision is made to store anything else which is a shame, but again, not too much of a limitation, and certainly no limitation at all if you are (like me) primarily a studio musician.

The MCS-1 disk program has a number of functions to perform which are: Upload and Download the sounds via BBC and MCS-1, to manage files, and perform all of the cursory BBC DFS operations such as rename and delete, to create special sound disks and test

CONTINUED OVER

MIDI information, to turn on or off codes and display the data it receives.

Simply !BOOTing the disk file will load the program, including a pretty MODE 2 picture. It is advisable to backup the disk. I would also advise taking the picture out of the loading sequence as it takes a little while to load in the program and delays are annoying.

The program is menu operated and is addressed by the function keys, as soon as the main program is loaded, it would be advisable to put a sound disk in and use the "Create" option, should you have a double sided double disk option, then you can create a sound disk on any drive. Users of 100K 40 Track had better be careful not to press F1 to F3 as this will attempt to read from a disk which does not exist and thus call up a "Drive fault" error.

After the disk is CREATEd, you should have files SOUND:1, SOUND:2 and SOUND:3, this is a special format disc and even though you can rename and delete files, you will have to do so through the program, as the disk is now unusable for storing anything else (BASIC programs for instance). You will find that as you experiment with more and more sounds, your disc requirements will treble at least, so get stocked up with good quality discs. Losing a program is bad enough, but I have found that losing a perfect sample of a vintage Stradivarius is almost as painful as watching a piano-wrecking contest.

Using the software, you can now accept MCS-1 data. In order to do this, you have to connect the MIDI interface to the MCS-1 by a length of cable connected to a 5 pin DIN plug, It is better if you either make the cables up (as I do myself for about 25p per meter) at considerable savings or buy a cassette to cassette copying lead from any Hi Fi shop. You would be mad to buy a Roland MIDI cable as the things are heat-sealed at the ends so as to make the cables impossible to repair in the event of an accident, and anyway, the ready-made and titled MIDI cables can be as much as £4.00 a pair!, not at all value for money!

Supposing you have a sampled sound in the MCS-1 which you wish to commit to the BBC's discs. You simply press F8 and selecting the requested file number, either 1 2 or 3, you then press the BBC button on the MCS-1. There is a startup message, if the connections are correct, telling you that the machines are in communication, **HELLO – MCS-1 VN1.1 HERE** . All this time there is a small graphics display of a "H" and a "+" symbol on the screen, to act as a prompt, telling you that the data is being passed, the DATA lamp on the MIDI interface will light up. Not only the sounds are stored though, thoughtfully all of the current VOICE parameters and LOOP parameters are loaded, so the digital delay and effects data are also available from a SAVEd file.

The first file, initially SOUND:1 will be used, if this is already full, then it will try SOUND:2 and so on until it passes SOUND:3, at which point it will give a "Disc Full" error. The RENAME option is useful so that SOUND:1 can be anything you want, providing the filename is no more than seven characters long.

Delete merely changes the file back to a SOUND file, if you made a mistake, then you can simply use RENAME, to repair the damage.

A very useful piece of software, built into the disc control program, is a MIDI test display page, called up by pressing F9. The keys F1 to F8 are toggles to set the transmit note ON/OFF codes for each MIDI channel. The MIDI receive section will likewise accept note on and off as well as display their binary value and position on the piano keyboard (great fun if you are using the Roland MIDI guitar controller!).

This sort of software is very useful if you have a slight problem remembering where transmission codes are going to, coming from and passing through what instruments. It can be a nightmare knowing which plug is which and if one breaks down on you, then you would normally have to try each channel. With the MIDI test software, it is possible to completely reorganise the connections and flow of data in software alone, the test software is invaluable if you are intending to go into MIDI in Poly mode, controlling 8 separate sound sources.

CONCLUSIONS

It is likely that the Powertran MCS-1 sound sampling unit is going to appeal to the studio engineer who has an existing MIDI arrangement. The advantages of housing sound sampling, digital delay and MIDI all under the same roof are obvious, the software is well implemented and is very flexible.

Sound sampling is however a two edged blade although it does allow the musician the creative ability to sample any sound they might want. The harsh samples *a la* Art of Noise and Frankie goes to Hollywood, are wearing a bit thin, but as sampling gets cheaper and more accessible (and with the MCS-1 how much more accessible can you get?) hopefully people with more inventive minds will put this process to better use. At the moment, all that sampling seems to be is a gimmick, a technological toy, which is a shame because sampling, is a branch of electronic music in itself, just as computer music is a branch of the wider world of contemporary music.

Sound samples are here to stay, the Powertran MCS-1 is a well designed, thorough piece of hardware. It is a little difficult to use in the beginning, because of the large number of switches, and the coarse, medium and fine control knob on the far right, but like any musical instrument, the more you use the MCS-1, the more you find out, and the more you want to use it.

For the professional user, I was pleasantly surprised how easily the MCS-1 fitted into the studio environment. Its rack mounting means that it can be fitted and tucked away permanently, but pulled out easily should the unit require additional cables in the back.

The unit was quite quiet in operation, although if it gets a little too close to high RF emitting pieces of equipment, then the unit can get a little noisy, so care must be taken when situating the unit in a studio.

The typical home computer musician is going to find the price of the MCS-1 expensive, although I think that the price of £499 for the Kit including all the parts (only advisable if you are *au fait* with a soldering iron) and £699 plus VAT for the assembled unit is enough to tempt the computer musician or the studio Engineer who is looking for a flexible and low cost sampling system to add to their MIDI keyboard. For more information, contact:

Powertran Cybernetics Ltd. Portway Industrial Estate, Andover, Hants, SP10 3NN.

SIEL COMPOSING SOFTWARE

For many musicians the idea of using a computer for recording music and storing sounds is a tempting alternative to recording on conventional tape. Although tape has its advantages, the big disadvantage is that tape costs a lot of money, especially when it is in the form of a 24 Track recording studio, normally costing up to £30 an hour.

In a studio the emphasis is on time and sometimes mistakes have to be left in the final mix. Unless you wish to go well overboard on the budget, you might find it easier, and more convenient, to record the sections at home, getting them all perfect and then taking the recording into the studio to play back your masterpiece.

Of course, not everybody would want to go into a studio to record their music. Perhaps all the musician wants is a convenient means with which to compose and play back at his leisure. Whichever way you look at it, the advantages of storing and recording the music at home and playing back, either in or out of the studio, far outweigh the disadvantages.

About five years ago, the Roland corporation came out with a "Micro Composer", which was basically a glorified sequencer, built into a dedicated computer.It used the Roland synchronising codes (which have now at last, bowed down due to the popularity of MIDI). You were able to play or "write" notes into the composer and you were able to play it back by plugging a Roland synthesiser into it.

Nowadays, the computer musician is demanding the same, if not more, features from a software package which can be loaded into a micro and, with the use of the software, control their equipment in much the same way that the old Micro Composers went about their business.

The Siel MIDI Composing software is the obvious step up from the typical one software package per computer rut. Their software is already implemented on the Spectrum and Commodore 64, and, at last on the BBC Micro. All of the software packages are hardware independent, hence the same interface and similar visual appearance of the software.

WHAT YOU GET

Both the software and hardware for the BBC micro are sold through Vince Hill Associates, who have a lot of experience with MIDI and computer music instruments. They act as consultants to the computer music world and we will no doubt see more of them as more professional computer software is released.

The Siel "Multitrack Composer and Arranger" is designed to work with the SIEL MIDI interface. They cannot guarantee that the software will work with other people's MIDI interfaces, but it is likely that you will be buying the software package and MIDI interface as a bundle together.

As usual, you can connect up to eight MIDI equipped instruments to the interface by daisy chaining them using the MIDI THRU sockets on the back of the equipment you are composing with. The program is totally menu driven and takes a little while to get used to.

THE HARDWARE

The Siel MIDI interface is one of the most stylish looking MIDI units I have seen. Its smart black and light blue livery, with the green LED at the front to say whether the unit is working or not, is appealing and simple. As opposed to many other interfaces, which only plug into the 1MHz bus, the Siel interface has to plug into the user port, probably to gain extra power for itself.

There are . three MIDI OUT sockets, which is useful to say the least. There is also the usual MIDI IN and THRU sockets and the ubiquitous IN CONTROL bus. There is a decent length of cable at the end, so it can be situated a little way away from the computer, if you need it to be elsewhere other than hugging the side of the BBC's case.

Plugging the unit in, is a simple case of sticking the two cables, which cannot be mixed up, into the plugs on the underside, one into the 1MHz bus and the other into the user port, once they are fitted, the BBC should be switched on. The green spy light should be on.

USING THE SOFTWARE

The composing software is on disc, and is loaded by !BOOTing the disk, (cassette based systems will be supported by the time you read this as the finishing touches to the cassette have to be made). A short loading sequence and you are put into the primary menu, which gives you three options: Composer, MIDI Control and Disk Operation. Pressing C,M or D will enter each corresponding second menu.

Pressing "M" will enter the MIDI control mode, this sets up the interface to transmit in one of the three MIDI '85 modes, OMNI, POLY or MONO.

OMNI mode is the very basic MIDI mode. When the synth receives this information it is instructed to play all of the notes sent by the computer, channel allocation (like calling one synth channel 1 and a drum unit 2,3 and 4 for instance) is ignored.

POLY mode is the more advanced form of transmit protocol which can send data down specific channels so a drum part can be sent down channels 2,3 and 4 to play a drum section. There is no conceivable reason why all 8 channels cannot be used on synths, or drum units!

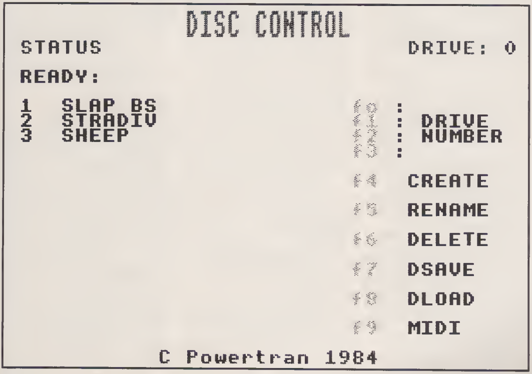
MONO mode allows you to use each channel of a synthesiser to control different sounds. Synths that can use this facility are, for example the Yamaha DX-7 and the SCI Six Trax and T8. Very useful if you wish to make the composing software handle the bulk of the recording.

The "Disk Operations" menu performs the functions you will need to LOAD SAVE, RENAME, DELETE and CLEAR files. Pressing LOAD, for instance clears the screen, and flashes the prompt: **Give Name of composition** . As it happens there is a demonstration file on the master disk, which is a section from Bach's Brandenberg Concerto, number 3.

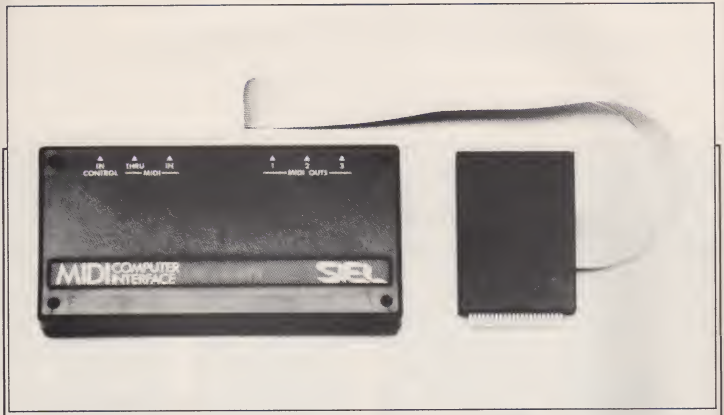
COMPOSING WITH SIEL

The manual I received was only a provisional one but all of the necessary information is safely contained within the 10 pages of A4. The most vital section of the software (and thank God the

CONTINUED OVER



```
                    DISC CONTROL
STATUS                                  DRIVE: 0

READY:

1   SLAP BS                        f0  :
2   STRADIU                        f1  :  DRIVE
3   SHEEP                          f2  :  NUMBER
                                   f3  :

                                   f4   CREATE

                                   f5   RENAME

                                   f6   DELETE

                                   f7   DSAVE

                                   f8   DLOAD

                                   f9   MIDI

              C Powertran 1984
```



manual didn't skimp on this part) is the composing language. There's a comprehensive section which leads the user by the hand into the complex programming language.

There is no similarity between the SIEL music composition language and AMPLE, the only other MCL on the BBC micro, and for those who have been brought up on the "sticks and blobs" of conventional music notation, I am afraid that you will find the process rather tedious. The SIEL MCL is, I would gather, designed for the Computer Musician, who is, generally speaking, happier with numbers to build up structures in music, somewhat like the structures built up through repetition and transposition, similar in the way Bach wrote some of his pieces, especially his preludes for solo Piano.

When you enter the menu, you will be greeted with the "Indication of State" (why does that sound like a funeral?) which is simply the computer informing you the state of MIDI channels 1 to 6. Figure 1 shows the column arrangement, the channel allocation, and the values of all of the stops and pauses. If you are composing music, your aim is to get the values in the last column to add up, if not, and you are not trying to do strange things with rhythm like Phillip Glass often does, then you have made an error. With care, the numbers should always add up.

The MCL has a surprisingly simple vocabulary but within the framework of the Siel composing software it is easy to build up quite a lot of music from surprisingly little code. All of the commands are single letters, for instance "C" is for COPY and "P" is for PLAY.

The syntax is very rigid and, in early days, your keying-in music will be fraught with mistakes. Just don't keep making the same mistake that I do, and that is press the RETURN key at the end of a line, this is fatal as it erases the music you have typed in, unless you have terminated the music with a "@" character at the end. Figure 2 is a typical example of how music is listed and put in the computer.

Should you be lucky enough to have a velocity sensitive synthesiser (the faster/harder you hit the keys, the louder the synthesiser plays back), then you will no doubt want to program this feature in. Using the Velocity sensitive option takes up no more space in composer memory than normal.

In the version of the software I used, the cursor keys are not properly implemented, so editing an erroneous line is impossible. Only re-typing the sequence gets the music into the computer. This is an area for improvement in future releases. The commands are listed in full in figure 3 with a brief description as to what they do.

INSERT is a useful facility if you wish to write velocity sensitive information. At a later date, should you wish to use velocity sensitive encoded data on an ordinary synth, then the code is ignored, (one of the beauties of MIDI is its ability to ignore things it doesn't understand). Using INSERT will re-number the other step numbers, in your current channel.

TRANSFER is useful if you wish to have another channel playing a certain instrument. For instance, if you had programmed a Bass section, and you wished to link the Kick drum of your MIDI equipped Drum unit (or computer!) with the Bass line of your score, then, it is a simple case of transferring the music laid down in the Bass section, to the channel allocated to the Kick Drum. Should you wish to transfer actual music scores but didn't want a different timbre, then it is a simple case of transposing the music up, say a third so the music will still fit and you can hear the effect of what you have done.

COPY is, again a very useful feature, allowing you to copy whole chunks of music. It is easy to write whole songs with the minimum of effort this way. You could write a "Song", using an introductory section, and a verse followed by a chorus, with a slightly changing Bass pattern every time.

NEW is used only when you want to start on a new composition, or you want to try afresh on a new channel, **(N),4** will clear channel four, but (N) will clear the whole lot!, very drastic, so SAVE the music file first.

AT WORK AND PLAY

It is wise to consider what kind of person is going to use this sort of hardware/software package. Certainly the computer musician will be happy with the facilities, especially for the price. The software boasts a true composition language dedicated to serving MIDI and works very well. I would imagine the greatest difficulties would arise for those who are used to actually playing an instrument to input their scores into the computer.

The Siel philosophy and approach to their software is novel to say the least. Their aim is, I think, to have all the peripheral instruments controlled from the computer. The software for the Commodore 64 to drive the Siel Expander 80 is clever, having an imaginary synthesiser front panel on the screen, to control an otherwise entirely digital synthesiser, with the addition of a MIDI keyboard a fully polyphonic synthesiser is built up at a fraction of the normal cost!

Siel's entry into computer music on the BBC Micro is most welcome. The Composing Software package and Midi Interface are an excellent start. I look forward to seeing what more Siel have in store.

Oxford Pascal

D.B.Stiles

BBC Pascal from the compiler factory of Oxford Computer Systems.

PASCAL has been extensively advertised recently as the panacea for programmers who are tired of BASIC. As a professional user of PASCAL on IBM machines I can vouch for the fact that it is more convenient in many ways, and fewer errors are encountered at run time than with BASIC. I chose the Oxford Computer Systems version of PASCAL for the BBC computer with a view to creating a much needed data-base for photographic slides.

The reasons for the choice of PASCAL over BASIC were twofold; the compiled code would (in theory) run faster on string searches, and PASCAL is more convenient and efficient in handling data-base records due to the inherent record structure of PASCAL. Whilst the former may or may not be true, the latter fell far short of expectations in this instance.

POSTAGE AND PACKING

Oxford PASCAL arrived in a plastic 'book', containing the manual, a ROM and a floppy disk. A cassette version is also available, but this is a rather more limited version which requires the ROM to always be present and permits only small programs to be compiled.

The disk version, on the other hand, has a disk-based compiler which allows large programs to be compiled directly onto disk. It also has a linker which allows very large programs to be compiled in separate smaller modules. These are generally tested separately as far as possible, and then linked into one complete working whole. It is also possible to include procedures and functions from previous programs by keeping them separately on disk as library files, then INCLUDE'ing them in the compilation.

A 'locate' facility is provided for either converting compiled code into stand-alone programs or for placing them at any memory location you may choose. Note the either/or! It is apparently not possible to relocate a program to run without the Oxford PASCAL ROM; thus a program could not be run, for example, on a cassette machine if it had been compiled on a disk-based machine, unless the ROM was in residence.

DOCUMENTATION

The Oxford manual is adequate if one is familiar with the language; however, it is not suitable as a beginners guide to the PASCAL language. More and better examples would be beneficial, even to the experienced PASCAL user. I would suggest something like Programming In Pascal by Peter Grogono as a tutorial for the tyro and a reference book for the experienced programmer.

The Oxford manual contains several errors, although most can be puzzled out with a bit of thought. The practice of using the underline character to separate words in a procedure name is supported, says the book. No it's not, says the compiler. The example showing a packed array of 15 characters being assigned. However nine characters will not even compile. The provision for simulating BASICS's OSCLI command line interpreter is also shown incorrectly, since it requires a writeln and the string in quotes.

Unlike my normal practice of 'try it first, then read the book', I actually read most of the manual before even plugging the ROM in. The first alarming thing I read in the manual was 'BASIC 2', since I only have BASIC 1. And the manual meant it too! I don't know why PASCAL should require BASIC at all, but the resident editor refused to work, and the sample programs, with one exception, refused to compile. The compiler seemed to fail them on real numbers at compile time, but the program then crashed at run time if the reals were replaced by integers.

The resident editor as supplied I can do without in any case, since it is a line editor similar to the BASIC one; although with the addition of find and replace facilities. As far as I am concerned one of the big attractions of PASCAL is that I can use a text editor with its facility for scrolling up and down through the program, and get away from the restrictions of a line editor. Even a word processor such as WORDWISE is preferable, and this is what I am currently using on PASCAL Of course, life is never smooth, and the Oxford PASCAL ROM does not live well with WORDWISE. In particular WORDWISE will no longer include a print file from disk in its presence.

FILING FAILING

On the whole these faults are mainly in the category of teething bugs, and seem to indicate too early a release of the product. If these were its only failings I could use it for several programs I have in mind. Indeed, I have written four smallish ones with it already. Its major faults are in not supplying adequate string handling, and providing only the basic file handling necessities.

There is no provision at all for random access files, and this means that my data-base ideas are just not on, as this type of application uses almost nothing else. A good random access filing system coupled with the record structure inherent in PASCAL makes it a language ideally suited to database management.

The Oxford implementation of PASCAL contains access points to the computer's operating system at several points, giving control over move, draw, sound etc. It does not, however, provide a direct equivalent of INKEY and the only way to check on the shift and control keys is by braving the wrath of Acorn and 'peeking' a memory location. There is no apparent way of defining constants for ASCII characters; allowing CONST chr(130) to define a function key constant would be very useful. Nor is it possible to do a SET on the full range of 0 to 255 for key checking, which in turn is necessary because CASE is lacking the ELSE function. Although non-standard, the ELSE or OTHERWISE option for CASE is almost universal in modern PASCAL implementations.

Now I know that string handling, OTHERWISE/ELSE and random access are not defined in the PASCAL standards, but any language which ignores them is doomed at birth for serious programming. Couple this with the fact that the I/O error trapping does not work on disk files and that ends its usefulness for general user-friendly applications. So much for the clause in the manual permitting the user of Oxford PASCAL to sell programs written in it.

After using Oxford PASCAL for a week I had compiled a short list of faults and comments. Not being able, for some reason, to raise them on the phone I wrote to Oxford enclosing my list. Apart from one abortive phone call from them which failed to reach me, I heard nothing from them for six weeks. When they did write, it was a brief note enclosed with a new (free) copy of version 2.1 disk and ROM, with a note to the effect that as far as they knew all the bugs had been removed, and that a new manual was in preparation.

My copy now seems to work with BASIC 1 quite happily, as evidenced by the fact that I can use the editor (admittedly useful for five-liners), and the sample programs now compile and run; but none of the other complaints seem to have been dealt with.

OXFORD SUMMARY

The Oxford implementation of PASCAL is a very good try on a machine with limited memory; although with disk machines this is not quite such a limitation as with cassette ones.

As a strict version of standard PASCAL the Oxford implementation is very reasonable, except for a few bugs (some of which may still be caused by BASIC 1). Thus it is a good introduction to higher level languages for students, providing they have a good teaching manual to aid them. As a serious working language, however, standard PASCAL itself is of little use. It is only when extensions such as disk and string handling are added to the language that it becomes the powerful tool with which real programs are written. ■